A Simple Reservoir Model of Working Memory with Real Values

Anthony Strock, Nicolas P. Rougier, Xavier Hinaut

A Simple Reservoir Model of Working Memory with Real Values

Anthony Strock[2,1,3], Nicolas Rougier[1,2,3], Xavier Hinaut[1,2,3]

1. Inria Bordeaux Sud-Ouest, Talence, France.
2. LaBRI, UMR 5800, CNRS, Bordeaux INP, Université de Bordeaux, Talence, France.
3. Institut des Maladies Neurodégénératives, UMR 5293, CNRS, Université de Bordeaux, Bordeaux, France.

MNEMOSYNE Team
first-name.last-name@inria.fr

Abstract

Prefrontal cortex is known to be involved in many high-level cognitive functions, in particular working memory. Here, we study to what extent a group of randomly connected units can store and maintain (as output) an arbitrary real value from a streamed input, *i.e.* how such system act as a sustained working memory module without being distracted by the input stream.

Furthermore, we explore to what extent such an architecture can take advantage of the stored value in order to produce non-linear computations. Systematic comparison between different architectures (with and without feedback, with and without a working memory unit) shows that explicit memory is required.

With Principal Component Analyses (PCA) we show that the reservoir state is encoding time and the memorized value in different ways depending if a supplementary task is required. Moreover, theses memory states are similar to attractors in an input-driven system [3], and in particular, similar to a noisy line attractor [6].

In this study, we did not try to find the optimal number of reservoir units needed for each task. Conversely, we voluntary limited the size of the reservoir to 100 neurons in order to see if such rather small reservoirs were sufficiently competitive.

Materials & Methods

Echo State Networks [1]

Update equation of the reservoir (recurrent layer) and the readout (output layer):

$$\mathbf{x}(t+1) = (1-\alpha)\mathbf{x}(t) + \alpha f(\mathbf{W}^{\mathbf{in}}\mathbf{u}(t+1) + \mathbf{W}\mathbf{x}(t)) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{W}^{\mathbf{out}}\mathbf{x}(t) \quad (2)$$

Matrices $\mathbf{W}^{\mathbf{in}}$ and $\mathbf{W}$ are randomly generated.

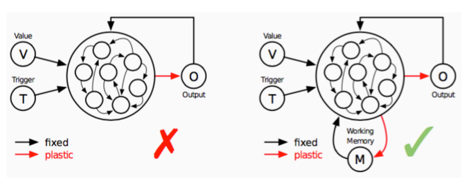
Training of the output weights with ridge regression

$$\mathbf{W}^{\mathbf{out}} = \mathbf{Y}^{\mathbf{d}}\mathbf{X}^{\mathbf{T}}(\mathbf{X}\mathbf{X}^{\mathbf{T}} + \beta\mathbf{I})^{-1} \quad (3)$$

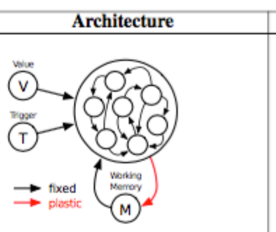
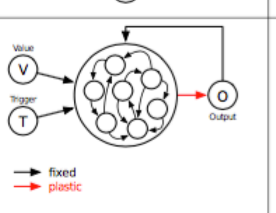
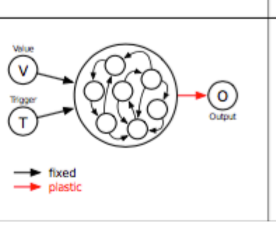
Results: Model comparison

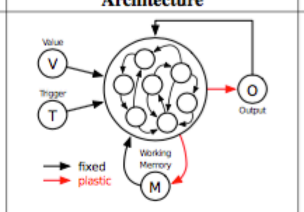
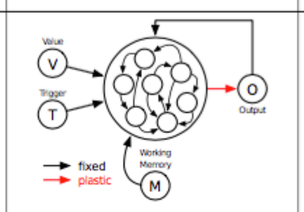
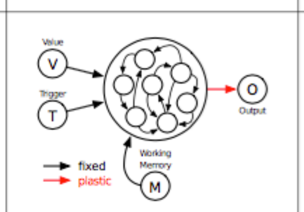
Main results for task 2 (RMSE)

No WM unit: 3.03e-1
With WM unit: 7.26e-4



Detailed results for tasks 1 & 2

Task	Architecture	RMSE
Memory only		**1.55e-4** ±7.42e-5
No explicit memory		3.03e-1 ±4.53e-4
No explicit memory (No FB)		3.05e-1 ±3.67e-4

Task	Architecture	RMSE
Trained explicit memory		**7.26e-4** ±1.88e-4
Oracle explicit memory		1.99e-4 ±3.15e-5
Oracle explicit memory (No FB)		7.10e-5 ±2.65e-5

Discussion

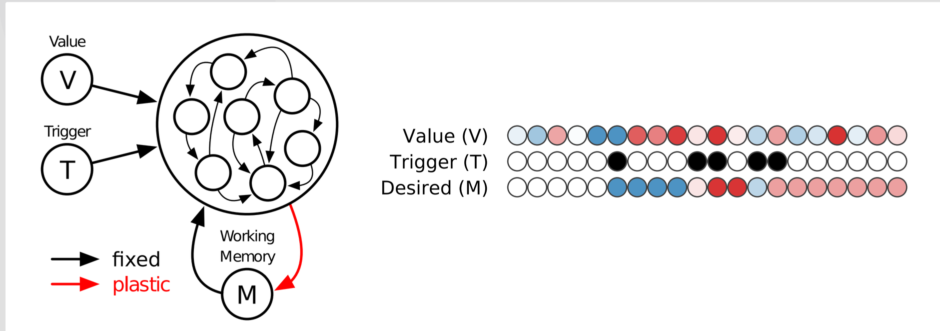
We have shown how a small group of randomly connected units is able to maintain an arbitrary value at an arbitrary time from a streamed input. It is to be noted that the model has not been trained at memorizing every possible value since there is virtually an infinite number of values between -1 and +1. What the model has actually learned is to gate an input value into a placeholder, a.k.a. a working memory (WM) unit. After training, this WM unit act like a gated memory: information enters while the gate is opened and is kept constant once the gate is closed. It can then be used to solve more complex tasks.

We have also demonstrated how such explicit working memory is critical in solving a scaling (i.e. multiplication) task. The same model deprived of the explicit working memory fails at solving the task and exhibits very bad performances (error of 3.03e-1 instead of 7.26e-4). This demonstrates the criticality of the presence of the working memory unit.

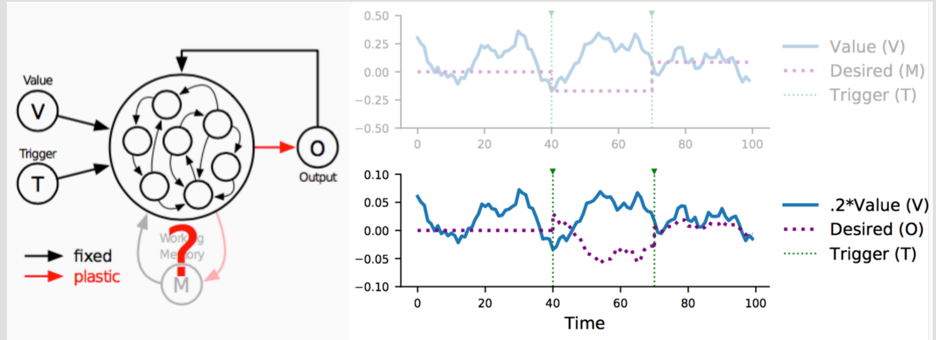
Future work will investigate similar mechanisms when online learning algorithms are used (*e.g.* reward-modulated rules [4] and FORCE learning variants [6, 7]), and investigate how temporal patterns such as *conceptors* [2] could be stored in WM units in a similar way.

Materials & Methods

Task 1: store value sync. with last trigger



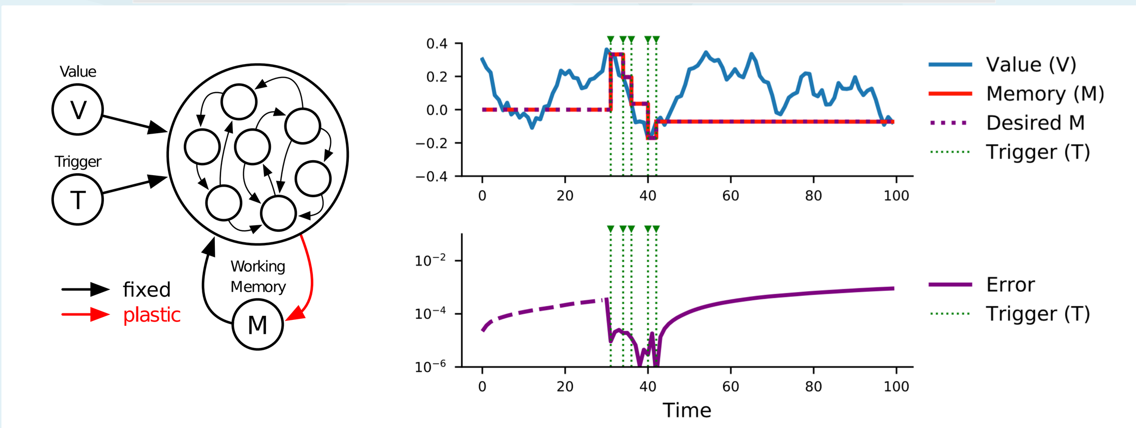
Task 2: scale input with last trigger sync. value



Global Results

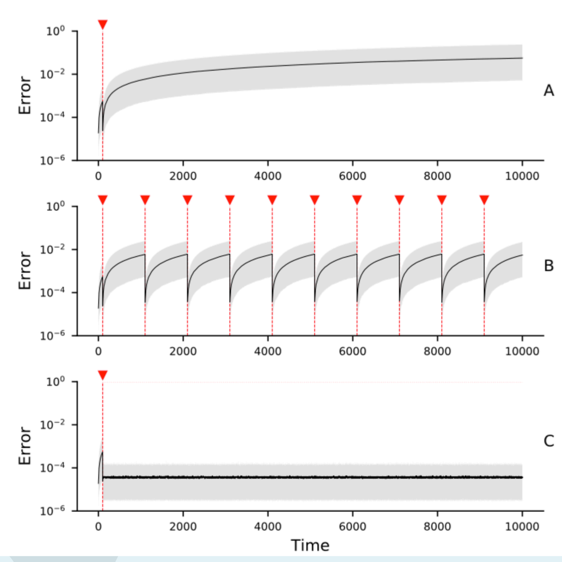
Can an arbitrary real value be maintained in a reservoir ?

Yes, and the value maintained is relatively stable



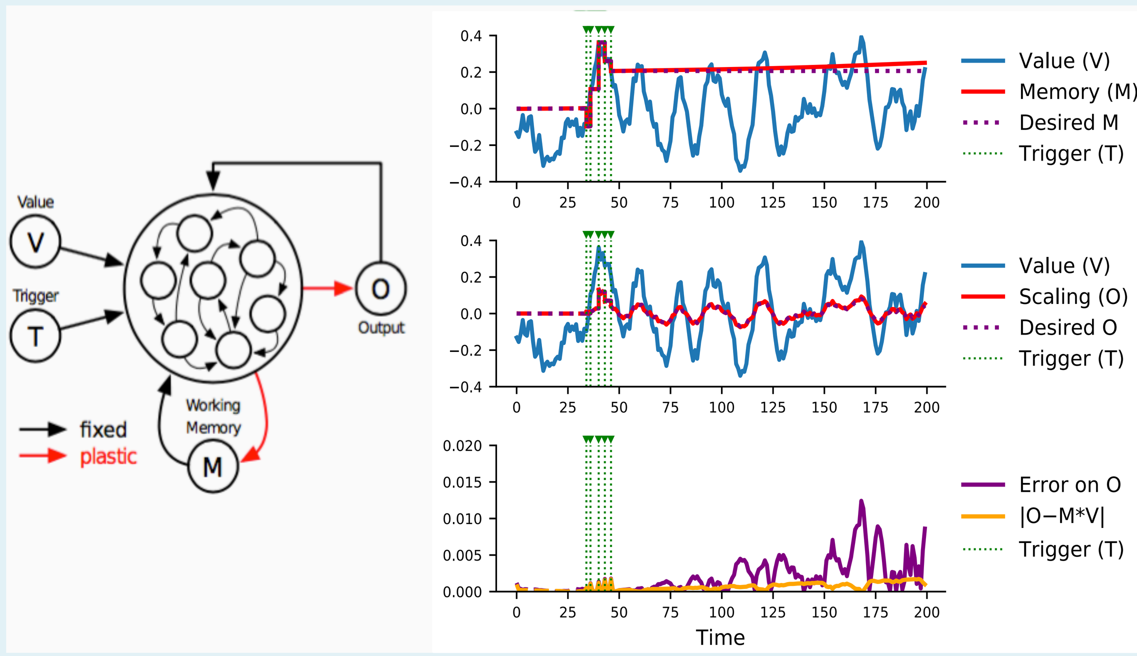
Testing the simple architecture for untrained scenarios

Scenario	RMSE (Mean±Std)
Training-like	2.28e-4±1.90e-4
Single (A)	4.55e-2±4.19e-2
Periodic (B)	5.04e-3±4.17e-3
Continuous (C)	9.25e-5±3.66e-5

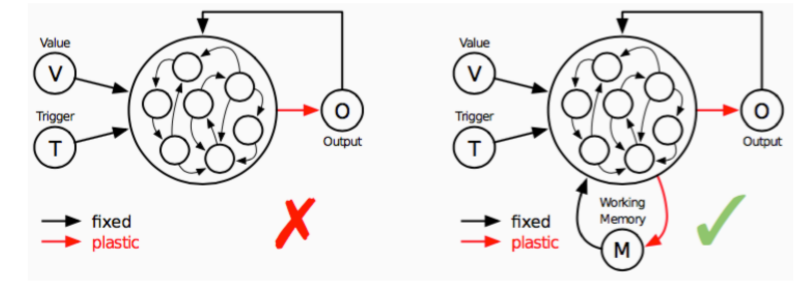


Does a reservoir need to explictly maintain values ?

Yes: no working memory unit → unable to solve the task

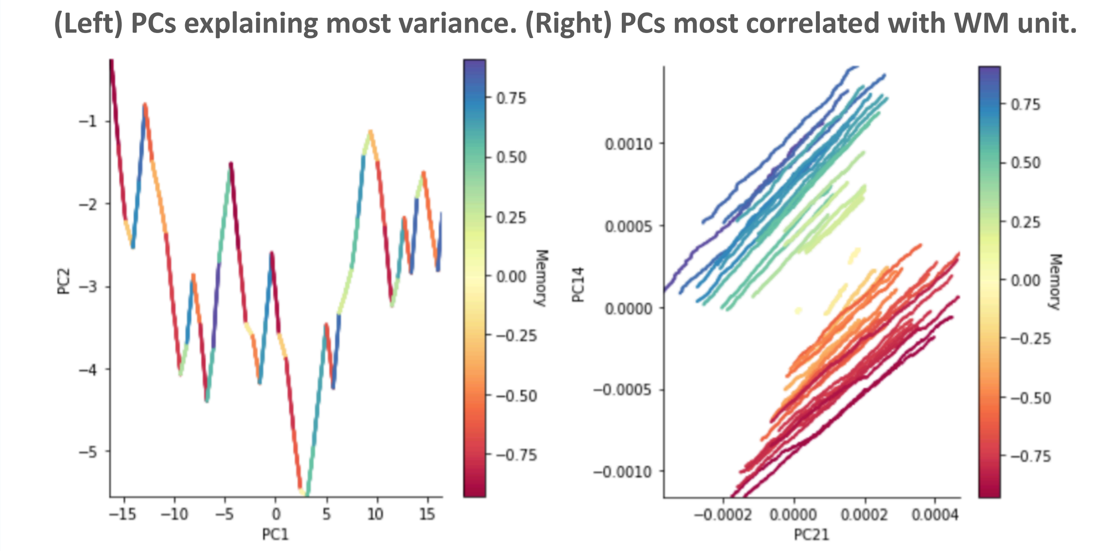


Need for WM unit (see model comparison panel for details)



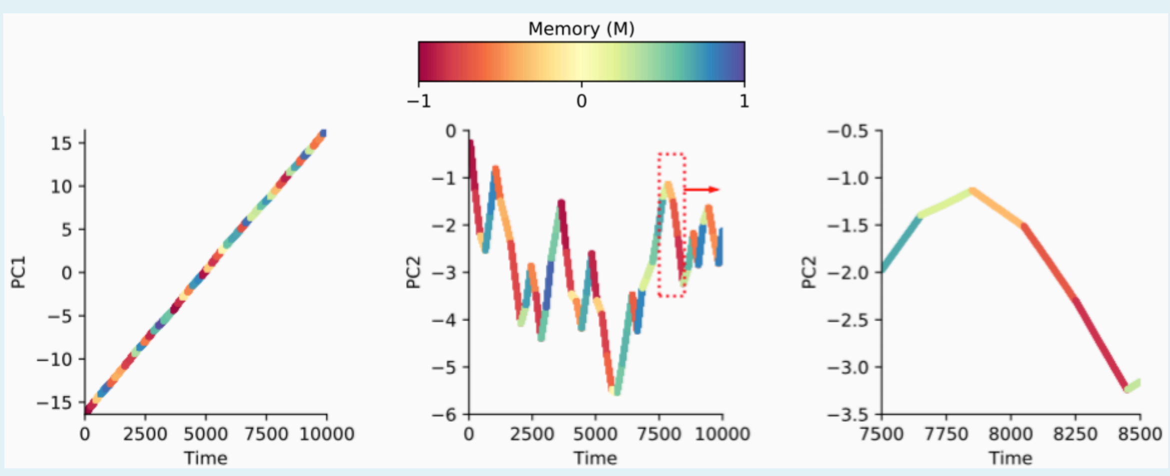
PCA analyses

(Left) PCs explaining most variance. (Right) PCs most correlated with WM unit.

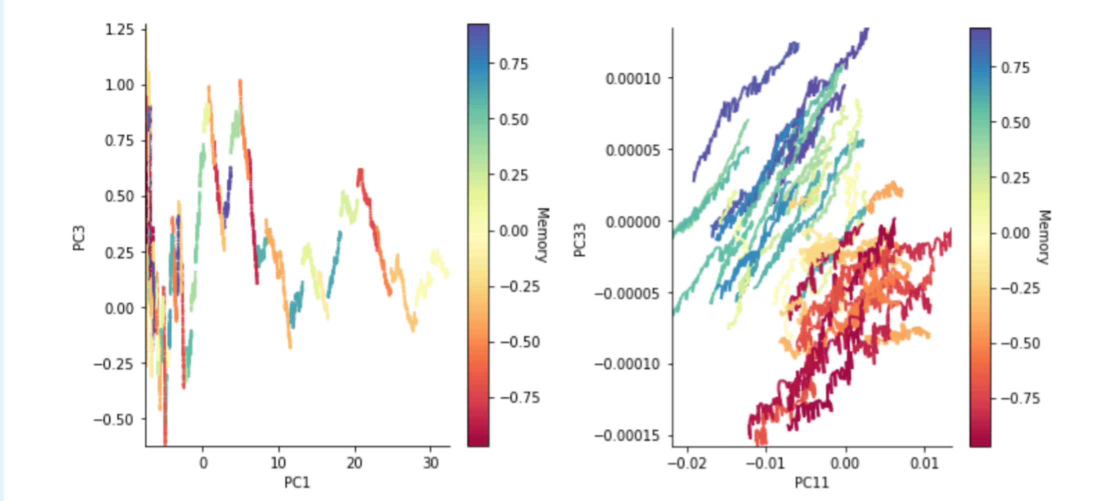


How is encoded the maintained value ?

- **PC1:** time linearly encoded
- **PC2:** memory encoded in the temporal derivative

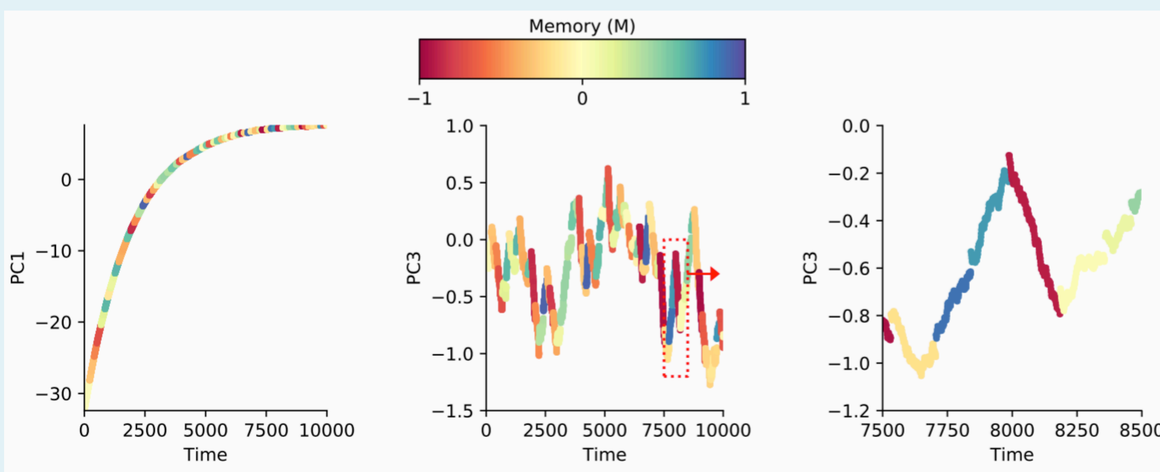


(Left) PCs explaining most variance. (Right) PCs most correlated with WM unit.



How is encoded the maintained value ?

- **PC1:** time non linearly encoded
- **PC3:** sign of the memory encoded in the temporal derivative



References

[1] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks" Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, vol. 148, p. 34, 2001.
[2] H. Jaeger. Controlling recurrent neural networks by conceptors. arXiv preprint arXiv:1403.3369, 2014.
[3] R Pascanu and H Jaeger. A neurodynamical model for working memory. Neural networks, 24(2):199–207, 2011.
[4] G M Hoerzer, R Legenstein, and W Maass. Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. Cerebral cortex, 24(3):677–690, 2012.
[5] Barak, O., & Tsodyks, M. (2014). Working models of working memory. Current opinion in neurobiology, 25, 20-24.
[6] Beer, C., & Barak, O. (2018). Dynamics of dynamics: following the formation of a line attractor. arXiv preprint arXiv:1805.09603.
[7] Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. Neuron, 63(4), 544-557.
[8] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in Proceedings of the 12th Python in Science Conference, pp. 13–20. 2013.
[9] P S Goldman-Rakic. Circuitry of primate prefrontal cortex and regulation of behavior by representational memory. Comprehensive Physiology, 1987.
[10] M Rigotti, O Barak, M R Warden, X-J Wang, N D Daw, E K Miller, and S Fusi. The importance of mixed selectivity in complex cognitive tasks. Nature, 497(7451): 585, 2013.
[11] H Jaeger. Long short-term memory in echo state networks: Details of a simulation study. Technical report, Jacobs University Bremen, 2012.
[12] J Dambre, D Verstraeten, B Schrauwen, and S Massar. Information processing capacity of dynamical systems. Scientific reports, 2:514, 2012.
[13] P F Dominey. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. Biological cybernetics, 73(3):265–274, 1995.

Links

Paper:
https://hal.inria.fr/hal-01803594v1
Source code:
https://github.com/anthony-strock/ijcnn2018

Related paper

Anthony Strock, Nicolas Rougier, Xavier Hinaut. A Simple Reservoir Model of Working Memory with Real Values. *IJCNN 2018 - International Joint Conference on Neural Networks*, Jul 2018, Rio de Janeiro, Brazil. 〈hal-01803594〉

université de BORDEAUX · CNRS · LaBRI · Inria INVENTEURS DU MONDE NUMÉRIQUE · imn institut des maladies neurodégénératives